Web Results **1** - **10** of about **145,000** for **hash** and **object** and **cache** and **invalidation**. (**0.31** seconds)

Inherent HTTP Coherence
Cache can calculate **object hash** upon storage, or first need. ... does not address higher-level issues such as freshness calculation vs. **invalidation**, ...
www.mnot.net/papers/coherence.html - 12k - Cached - Similar pages

Re: MD5 **hash** query from Joe Cooper on 2001-12-28 (squid-dev)
requests to Squid to **invalidate** the **objects** that need to be invalidated. The reason I need the **hash** is because in order to provide "full tree" ...
www.squid-**cache**.org/mail-archive/squid-dev/200112/0150.html - 12k - Cached - Similar pages

MANUAL.rd
The class **Cache** looks like a variant of **Hash** , and, in fact, **Cache** supports most of the methods of **Hash** . To control how to **invalidate** excessive **objects**, ...
www.nongnu.org/pupa/ruby-**cache**-MANUAL.html - 10k - Cached - Similar pages

[PDF] A Hierarchical Internet **Object Cache**
File Format: PDF/Adobe Acrobat - View as HTML
Objects in the **cache** are referenced via a **hash** table keyed ... archical **invalidation** schemes, but that sites exporting time- ...
www.cs.princeton.edu/courses/archive/fall03/cs518/papers/harvest.pdf - Similar pages

[PDF] **Cache Invalidation** and Update Propagation in Distributed Caches ...
File Format: PDF/Adobe Acrobat - View as HTML
an **object** hits the **cache**. In server-side **invalidation**, ... **cache** server incorporates an **object cache**,. which is implemented as a Java **Hash** Table data struc- ...
www4.ncsu.edu/~rychirko/Papers/KohliC05.pdf - Similar pages

[PDF] Functional Specification for **Object Caching** Service for Java ...
File Format: PDF/Adobe Acrobat - View as HTML
2x) of retrieving from a **hash** table. Ideally it would be nice to be as ... Flushing the memory **cache** will also **invalidate** memory **objects** spooled to disk. ...
jcp.org/aboutJava/communityprocess/jsr/**cache**FS.pdf - Similar pages

Core API Documentation: Class CacheTable
If the **hash** code sent by node X matches the **hash** code of the **object** mapped to the same key on node Y, then the **cache** entry maintained by Y need not be ...
byline.**objectweb**.org/ccm-core-6.1.0/api/com/arsdigita/caching/**Cache**Table.html - 28k - Cached - Similar pages

Object-Oriented DBMS
Goal: Allow clients to **cache objects**/pages and maybe locks from server for ... Mistaken **invalidation** is managed by keeping a LIFO list of **invalidate** page ...
redbook.cs.berkeley.edu/redbook3/lec26.html - 16k - Cached - Similar pages

WebSphere Dynamic **Cache**: Improving J2EE application performance
A **cache** policy using the **hash** of the SOAP envelope could be defined as follows: ... Table 6 Performance measurements using **object cache and invalidation** ...
www.research.ibm.com/journal/sj/432/bakalova.html - 89k - Cached - Similar pages

Sign in

Google

Web Images Groups News Froogle Maps more »

hash and key and cache and invalidation Search Advanced Search Preferences

The "AND" operator is unnecessary -- we include all search terms by default. [details]

Web Results **1** - **10** of about **147,000** for **hash** and **key** and **cache** and **invalidation**. (**0.23** seconds)

Tip: Save time by hitting the return key instead of clicking on "search"

Re: MD5 **hash** query from Joe Cooper on 2001-12-28 (squid-dev)
clean why I need to be able to match the **hash**:. I'm making a utility that will provide **invalidation** services for Squid. The first is simply an indexer that ...
www.squid-**cache**.org/mail-archive/squid-dev/200112/0150.html - 12k - Cached - Similar pages

Encrypted **key cache** - Patent 6981138
with the **hash** value as the **key**, or Eh(F)(F). Next, read access control ... For example, a **cache** entry **invalidation** command or message could be sent by some ...
www.freepatentsonline.com/6981138.html - 83k - Cached - Similar pages

MANUAL.rd
The class **Cache** looks like a variant of **Hash** , and, in fact, **Cache** supports most of the ... Return keys and cached objects as a **Hash** object. **invalidate(key** ...
www.nongnu.org/pupa/ruby-**cache**-MANUAL.html - 10k - Cached - Similar pages

'RE: External/Event Based **Cache Invalidation** (somewhat long)' - MARC
When the **key** is ejected from the **cache** for other > reasons (another > > > pipeline ... because of > > grouping of multiple keys under the same **hash** code. ...
marc2.theaimsgroup.com/?l=xml-cocoon-dev&m=105698867207856&w=2 - Similar pages

> 'RE: External/Event Based **Cache Invalidation** (somewhat long)' - MARC
> When the **key** is ejected from the **cache** for other reasons (another > pipeline ... In order for this to work, all keys must return the exact same **hash** code. ...
> marc2.theaimsgroup.com/?l=xml-cocoon-dev&m=105688316717880&w=2 - Similar pages

Controlling **Cache** Behaviour for Portlet Markup Fragments in ...
Enhanced **cache key** Enhanced **cache key**. The state **hash** can be used as a shorthand ... Finally, **cache invalidation** can also be done programmatically. ...
www.ibm.com/developerworks/websphere/library/techarticles/0401_voth/0401_voth.html - 67k - Cached - Similar pages

[PS] Tasks and Revocation for Java (or, Hey! You got your Operating ...
File Format: Adobe PostScript - View as HTML
For instance, a **hash** table object expecting keys oftype " ... Luckily, standard **caching** and **invalidation** techniques apply to remote pointer accesses, ...
www.cs.cornell.edu/Info/People/hawblitz/PLDI2000-submit/luna-99-11-13.ps - Similar pages

Distributed **Caching** with Memcached | Linux Journal
A **hash** function is a one-way function mapping a **key** (be it numeric or ... This is unfortunate when the write is an important **cache invalidation** operation. ...
www.linuxjournal.com/article/7451 - 54k - Aug 1, 2006 - Cached - Similar pages

Core API Documentation: Class CacheTable
If the **hash** code sent by node X matches the **hash** code of the object mapped to the same **key** on node Y, then the **cache** entry maintained by Y need not be ...
byline.objectweb.org/ccm-core-6.1.0/api/com/arsdigita/caching/**Cache**Table.html - 28k -

The recent database difficulties have been resolved. Please let us know if you encounter any data corruptions.

CiteSeer Find: hashing and cache and invalidation Documents Citations

Searching for **hashing and cache and invalidation**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP
6 documents found. **Order: number of citations.**

A Scalable Approach to Thread-Level Speculation - Steffan, Colohan, Zhai, Mowry (2000) (Correct) (32 citations)
loop in Figure 1(a) which accesses elements in a **hash** table. This loop cannot be statically parallelized extension of writeback **invalidation**-based **cache** coherence (which itself scales both up and down)
www.cs.cmu.edu/~steffan/items/isca00.ps

One or more of the query terms is very common - only partial results have been returned. Try Google (CiteSeer).

Architectural Support for Thread-Level Data Speculation - Steffan, Colohan, Mowry (1997) (Correct) (11 citations)
loop in Figure 1(a) which accesses elements in a **hash** table. This loop cannot be statically parallelized through an extension to **invalidation**based **cache** coherence: i) detecting dependence violations,
reports-archive.adm.cs.cmu.edu/anon/1997/CMU-CS-97-188.ps.gz

Odin: Implications and Performance of a Novel DSM Design - Pears (1996) (Correct) (6 citations)
in particular memory modules. The resulting **hashing** effect on network traffic reduces access of locality and data caching, and memory and **cache** consistency. The contributions of the paper lie instructions. This protocol eliminates all **invalidation** traffic within the machine. To ensure correct
www.cs.latrobe.edu.au/~pears/PAPERS/96/icse.ps

Approaches for Broadcasting Temporal Data in Mobile Computing .. - Lam, Chan, Yuen (Correct)
www.cs.cityu.edu.hk/~rtmm/pub/mcs_jp1.pdf

Extending Cache Coherence to Support Thread-Level Data.. - Steffan, Colohan, Mowry (1998) (Correct)
of Systems while(continue cond) f x =**hash**[index1]**hash**[index2] y g Epoch 1
Extending **Cache** Coherence to Support Thread-Level Data
reports-archive.adm.cs.cmu.edu/anon/1998/CMU-CS-98-171.ps

Reducing Controller Contention in Shared-Memory.. - Talbot, Kelly (Correct)
is used to reduce contention. We present a **hashing**-based proxy placement scheme, and evaluate a phjk@doc.ic.ac.uk Abstract In simple **cache** coherency protocols, serialisation can occur when lines of which a copy is held in this **cache** (**invalidations** and replacements)and lines whose "home" is
www-ala.doc.ic.ac.uk/~samt/improving_tech.letter.ps.gz

Try your query at: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP

The recent database difficulties have been resolved. Please let us know if you encounter any data corruptions.

CiteSeer Find: invalidation and cache and hashing Documents Citations

Searching for **invalidation and cache and hashing**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP
6 documents found. **Order: number of citations.**

A Scalable Approach to Thread-Level Speculation - Steffan, Colohan, Zhai, Mowry (2000) (Correct) (32 citations)
it is a straightforward extension of writeback **invalidation**-based **cache** coherence (which itself scales
extension of writeback **invalidation**-based **cache** coherence (which itself scales both up and down)
www.cs.cmu.edu/~steffan/items/isca00.ps

One or more of the query terms is very common - only partial results have been returned. Try Google (CiteSeer).

Architectural Support for Thread-Level Data Speculation - Steffan, Colohan, Mowry (1997) (Correct) (11 citations)
pieces of functionality through an extension to **invalidation**based **cache** coherence: i) detecting dependence
through an extension to **invalidation**based **cache** coherence: i) detecting dependence violations,
reports-archive.adm.cs.cmu.edu/anon/1997/CMU-CS-97-188.ps.gz

Odin: Implications and Performance of a Novel DSM Design - Pears (1996) (Correct) (6 citations)
instructions. This protocol eliminates all **invalidation** traffic within the machine. To ensure correct
of locality and data caching, and memory and **cache** consistency. The contributions of the paper lie
in particular memory modules. The resulting **hashing** effect on network traffic reduces access
www.cs.latrobe.edu.au/~pears/PAPERS/96/icse.ps

Approaches for Broadcasting Temporal Data in Mobile Computing .. - Lam, Chan, Yuen (Correct)
www.cs.cityu.edu.hk/~rtmm/pub/mcs_jp1.pdf

Extending Cache Coherence to Support Thread-Level Data.. - Steffan, Colohan, Mowry (1998) (Correct)
this functionality we present an extension to **invalidation**-based **cache** coherence which is both scalable
Extending **Cache** Coherence to Support Thread-Level Data
reports-archive.adm.cs.cmu.edu/anon/1998/CMU-CS-98-171.ps

Reducing Controller Contention in Shared-Memory.. - Talbot, Kelly (Correct)
lines of which a copy is held in this **cache** (**invalidations** and replacements)and lines whose "home" is
phjk@doc.ic.ac.uk Abstract In simple **cache** coherency protocols, serialisation can occur when
is used to reduce contention. We present a **hashing**-based proxy placement scheme, and evaluate a
www-ala.doc.ic.ac.uk/~samt/improving_tech.letter.ps.gz

Try your query at: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP

Terms used **cache** and **invalidation** and **hashing** Found **6,375** of **182,582**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Client-side caching for TLS

Hovav Shacham, Dan Boneh, Eric Rescorla

November 2004 **ACM Transactions on Information and System Security (TISSEC)**, Volume 7 Issue 4

Publisher: ACM Press

Full text available: pdf(182.01 KB) Additional Information: full citation, abstract, references, index terms

We propose two new mechanisms for caching handshake information on TLS clients. The "fast-track" mechanism provides a client-side cache of a server's public parameters and negotiated parameters in the course of an initial, enabling handshake. These parameters need not be resent on subsequent handshakes. Fast-track reduces both network traffic and the number of round trips, and requires no additional server state. These savings are most useful in high-latency environments such as wireless network ...

Keywords: Bloom filters, TLS, session cache, wireless networks

2 Session 8: distributed systems: Transparent caching with strong consistency in dynamic content web sites

Cristiana Amza, Gokul Soundararajan, Emmanuel Cecchet

June 2005 **Proceedings of the 19th annual international conference on Supercomputing ICS '05**

Publisher: ACM Press

Full text available: pdf(370.07 KB) Additional Information: full citation, abstract, references

We consider a cluster architecture in which dynamic content is generated by a database back-end and a collection of Web and application server front-ends. We study the effect of transparent query caching on the performance of such a cluster. Transparency requires that cached entries be invalidated as a result of writes. We start with a coarse-grain table-level automatic invalidation cache. Based on observed workload characteristics, we enhance the cache with the necessary dependency tracking and ...

3 Energy-efficient selective cache invalidation

Jun Cai, Kian-Lee Tan

December 1999 **Wireless Networks**, Volume 5 Issue 6

Publisher: Kluwer Academic Publishers

Full text available: pdf(201.67 KB) Additional Information: full citation, references, citings, index terms